



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/765,882	01/16/2001	Loay Abu-Hussein	10992693-1	9296

7590 08/13/2004

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort collins, CO 80527-2400

EXAMINER

ZHOU, TING

ART UNIT

PAPER NUMBER

2173

DATE MAILED: 08/13/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/765,882	ABU-HUSEIN, LOAY
Examiner	Art Unit	
Ting Zhou	2173	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 28 April 2004.

2a) This action is **FINAL**. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1,2,6-14 and 16-22 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) 1-2, 6-14, and 16-22 is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on 28 April 2004 is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All b) Some * c) None of:

1. Certified copies of the priority documents have been received.

2. Certified copies of the priority documents have been received in Application No. _____.

3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)

2) Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) Notice of Informal Patent Application (PTO-152)

6) Other: _____.

DETAILED ACTION

1. The amendment filed on 28 April 2004 have been received and entered. Of the original claims 1-20, claims 3-5 and 15 have been cancelled by the applicant and are therefore withdrawn from consideration; the applicant has further added new claims 21 and 22. Therefore, claims 1-2, 6-14 and 16-22 as amended are pending in the application.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1-2, 6-14 and 16-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Yan et al. U.S. Patent 6,003,065 and Inoue et al. U.S. Patent 6,456,388.

Referring to claim 1, Yan et al. teach an appliance application loading system for a network environment (Yan et al.: column 2, lines 54-60). Specifically, Yan et al. teach a client (a thin client via the peripheral device) (Yan et al.: column 19, lines 38-40), a web application server communicating with the client within the network environment (a host computer communicating with the peripheral devices over the network) (Yan et al.: column 6, lines 52-67), a network-based appliance (peripheral device) communicably attached with the web server within the network environment (the host computer communicates with the peripheral device via communication links such as the Internet (Yan et al.: column 6, lines 52-67), wherein the

network-based appliance includes an embedded device having a non-volatile storage device (virtual machine instruction processors embedded in the peripheral devices) (Yan et al.: column 7, lines 4-8 and 38-45, column 23, lines 53-56 and column 197, lines 14-18), and a loading mechanism (information downloading mechanism) provided on the network-based appliance and operative to download an application to the appliance from the web application server upon the occurrence of a power on/off cycle (power on self test which upon boot up time, downloads the functions and applications to the peripheral device) (Yan et al.: column 20, lines 63-67 through column 21, lines 1-14 and 23-29), wherein a universal resource locator (URL) is stored on the non-volatile storage device, and an application body is provided on the web server at a location corresponding with the URL, the URL being initialized to access the application body (the URL of the application to be downloaded from the web server can be downloaded to reside in the peripheral device's storage while the server holds the material of the application to be downloaded) (Yan et al.: column 11, lines 21-24 and column 16, lines 20-24). This is further shown in Figures 1 and 2. However, although Yan et al. teach a URL stored on the non-volatile storage device, Yan et al. fail to explicitly teach an application header stored on the non-volatile storage device, the URL being initialized to access the application body when the loading mechanism detects the application header. Inoue et al. teach a system for loading an application onto a peripheral device from a connected computer network (Inoue et al.: column 2, lines 21-25) similar to that of Yan et al. In addition, Inoue et al. further teach an application header stored on the non-volatile storage device, the URL being initialized to access the application body when the loading mechanism detects the application header (the data file includes a header, where the application information is stored and upon receipt of the data file and based on the header

information in the data file, the application from the application server can be downloaded) (Inoue et al.: column 7, lines 1-8 and column 8, lines 10-16). It would have been obvious to one of ordinary skill in the art, having the teachings of Yan et al. and Inoue et al. before him at the time the invention was made, to modify the application loading system taught by Yan et al. to include the application header of Inoue et al. One would have been motivated to make such a combination in order to reduce the burden on a client computer by saving the time and money needed in developing the high capacity of a hard disk or a memory that would have otherwise been necessary in order to continuously store an entire application; by using an application header to store information and downloading the application only according to the header, the space, memory and management of the client device can be conserved and therefore, the processing and operation efficiency can be improved.

Referring to claim 2, Yan et al. teach the loading mechanism provided at least in part by the client (the loading of the function and applications occurs in the peripheral device, which is a thin client) (column 19, lines 38-43 and column 21, lines 23-29).

Referring to claim 6, Yan et al. teach the application body comprising a servlet provided on the web server, as recited in column 5, lines 40-67. According to the definition provided by Hyper Dictionary (<http://www.hyperdictionary.com>), a servlet is “a Java program that runs as part of a network service, typically an HTTP server and responds to requests from clients. The most common use for a servlet is to extend a web server by generating web content dynamically. For example, a client may need information from a database; a servlet can be written that receives the request, gets and processes the data as needed by the client and then returns the result to the client.” Yan et al. teach a network (host computer) that responds to requests from

clients. The peripheral devices requests information (applications) from the database and the virtual machines instruction processors process the information and return the results to the peripheral devices.

Referring to claim 7, Yan et al. teach the network-based appliance comprising an embedded device and the loading mechanism comprising a virtual machine (the peripheral devices comprises embedded virtual machine instruction processors and the downloading mechanism of the peripheral devices comprises a virtual machine), as recited in column 5, lines 60-67 and column 7, lines 4-8.

Referring to claim 8, Yan et al. teach the network-based appliance (peripheral device) using the loading device to download specific appliance configuration settings, as recited in column 14, lines 38-40 and column 21, lines 12-17.

Referring to claim 9, Yan et al. teach the appliance comprising an embedded device, and the loading mechanism comprising a program routine that copies an application program into the memory of the embedded device from the web server for execution (downloading functions associated with the applications into the peripheral device), as recited in column 21, lines 23-29 and column 23, lines 53-59.

Referring to claim 10, Yan et al. teach a computer peripheral program product comprising a web application server, a network environment, a computer peripheral and a application loader to load an extendable architecture application to the computer peripheral (downloading applications and functions to a peripheral device) so as to enable versioning, updating and remote configuration of the computer peripheral (Yan et al.: column 23, lines 47-50 and column 24, lines 7-9) via the web application server, wherein the application loader

associates an application body of the web application server (the URL header of the application to be downloaded from the web server can be downloaded to reside in the peripheral device's storage while the server holds the material of the application to be downloaded) (Yan et al.: column 11, lines 21-24 and column 16, lines 20-24), the application includes a uniform resource locator (URL) to the application , the application body including one or more individual applications that can be loaded on the computer peripheral (more than one document can be downloaded to the peripheral device) (Yan et al.: column 11, lines 21-24 and column 16, lines 20-24), the URL being initialized to access the application body (the URL of the application to be downloaded from the web server can be downloaded to reside in the peripheral device's storage while the server holds the material of the application to be downloaded) (Yan et al.: column 11, lines 21-24 and column 16, lines 20-24). This is further shown in Figures 1 and 2. However, Yan et al. fail to explicitly teach an application header having identification information for the application, the URL being initialized to access the application body when the loading mechanism detects the application header. Inoue et al. teach a system for loading an application onto a peripheral device from a connected computer network (Inoue et al.: column 2, lines 21-25) similar to that of Yan et al. In addition, Inoue et al. further teach an application header having identification information for the application, the URL being initialized to access the application body when the loading mechanism detects the application header (the data file includes a header, where the application information is stored and upon receipt of the data file and based on the header information in the data file, the application from the application server can be downloaded) (Inoue et al.: column 7, lines 1-8 and column 8, lines 10-16). It would have been obvious to one of ordinary skill in the art, having the teachings of Yan et al. and Inoue et al.

before him at the time the invention was made, to modify the application loading system taught by Yan et al. to include the application header of Inoue et al. One would have been motivated to make such a combination in order to reduce the burden on a client computer by saving the time and money needed in developing the high capacity of a hard disk or a memory that would have otherwise been necessary in order to continuously store an entire application; by using an application header to store information and downloading the application only according to the header, the space, memory and management of the client device can be conserved and therefore, the processing and operation efficiency can be improved.

Referring to claim 11, Yan et al. teach the computer peripheral comprises a virtual machine including a web client (thin client), as recited in column 7, lines 38-45 and column 19, lines 38-40.

Referring to claim 12, Yan et al. teach the computer peripheral (peripheral device) comprising a printer (as shown by reference character “102B” in Figure 10), and updating comprising configuring the printer with a printer application comprising a printer configuration state, as recited in column 21, lines 12-17 and column 23, lines 58-65.

Referring to claim 13, Yan et al. teach the printer configuration state comprising user settings (user selects a specific operation to perform and the system configures and sets the device), as recited in column 21, lines 12-17.

Referring to claim 14, Yan et al. teach a servlet on the web application server that transfers applications and settings to the printer in response to a power cycle that automatically updates the application and configuration settings for the printer, as recited in column 20, lines 63-67 and column 21, lines 1-5.

Referring to claim 16, Yan et al. teach, in column 5, lines 40-67, column 6, lines 52-67 and column 21, lines 1-29, a method for updating applications to embedded devices (downloading applications to peripheral devices), comprising providing a network-based appliance (peripheral devices such as printers) communicably attached with a web application server, the appliance having a loading mechanism to download an application to the appliance from the server, updating the appliance with the application from the server upon the occurrence of a power on/off cycle (power on self test which upon boot up time, downloads the functions and applications to the peripheral device) (Yan et al.: column 20, lines 63-67 through column 21, lines 1-14 and 23-29). However, Yan et al. fail to explicitly teach querying the appliance with the web server to determine presence of an application header and updating the appliance with the application upon detecting the presence of the application header. Inoue et al. teach a system for loading an application onto a peripheral device from a connected computer network (Inoue et al.: column 2, lines 21-25) similar to that of Yan et al. In addition, Inoue et al. further teach querying the appliance with the web server to determine presence of an application header and updating the appliance with the application upon detecting the presence of the application header (the data file includes a header, where the application information is stored and upon receipt of the data file and checking the header, the application from the application server can be downloaded based on the header information in the data file) (Inoue et al.: column 7, lines 1-8 and column 8, lines 10-16). It would have been obvious to one of ordinary skill in the art, having the teachings of Yan et al. and Inoue et al. before him at the time the invention was made, to modify the application loading system taught by Yan et al. to include the application header of Inoue et al. One would have been motivated to make such a combination in order to reduce the

burden on a client computer by saving the time and money needed in developing the high capacity of a hard disk or a memory that would have otherwise been necessary in order to continuously store an entire application; by using an application header to store information and downloading the application only according to the header, the space, memory and management of the client device can be conserved and therefore, the processing and operation efficiency can be improved.

Referring to claim 17, Yan et al. teach the appliance comprising an embedded device, and updating comprising configuring the embedded device with an application (downloading an application) comprising an embedded device configuration state, as recited in column 21, lines 12-29.

Referring to claim 18, Yan et al. teach the embedded device configuration state comprising user settings, as recited in column 21, lines 12-17.

Referring to claim 19, Yan et al. teach a servlet on the web application server that is transferred to the embedded device in response to a power cycle that automatically updates the application and configuration settings for the embedded device, as recited in column 20, lines 63-67 and column 21, lines 1-5.

Referring to claim 20, Yan et al. teach a plurality of appliances communicably attached with the web application server (as shown in Figure 1) each with a dedicated one of the loading mechanism, wherein the web application server stores appliance applications and configuration settings (column 20, lines 63-67 and continuing onto column 21, lines 1-29) to enable plural appliance configuration setup to version and update such applications (column 23, lines 47-50 and column 24, lines 7-9).

Referring to claim 21, Yan et al. teach an appliance application loading system for a network environment (Yan et al.: column 2, lines 54-60). Specifically, Yan et al. teach a client (a thin client via the peripheral device) (Yan et al.: column 19, lines 38-40), a server communicating with the client (a host computer communicating with the peripheral devices over the network) (Yan et al.: column 6, lines 52-67), a network-based appliance (peripheral device) communicably attached with the web server within the network environment (the host computer communicates with the peripheral device via communication links such as the Internet (Yan et al.: column 6, lines 52-67), wherein the network-based appliance includes an embedded device having a non-volatile storage device (virtual machine instruction processors embedded in the peripheral devices) (Yan et al.: column 7, lines 4-8 and 38-45, column 23, lines 53-56 and column 197, lines 14-18), and a loading mechanism (information downloading mechanism) provided on the network-based appliance and operative to download an application to the appliance from the web application server upon the occurrence of a power on/off cycle (power on self test which upon boot up time, downloads the functions and applications to the peripheral device) (Yan et al.: column 20, lines 63-67 through column 21, lines 1-14 and 23-29), wherein a universal resource locator (URL) is stored on the non-volatile storage device, and an application body is provided on the web server at a location corresponding with the URL, the URL being initialized to access the application body (the URL of the application to be downloaded from the web server can be downloaded to reside in the peripheral device's storage while the server holds the material of the application to be downloaded) (Yan et al.: column 11, lines 21-24 and column 16, lines 20-24), wherein servlet settings corresponding to the network-based appliance are automatically updated via the loading mechanism if a user locally changes settings of the

network based appliance (when the peripheral device has, upon user initiation, upgraded or updated its settings, the peripheral device automatically sends a notification applet to the host computer notifying the host that the peripheral device has been changed) (Yan et al.: column 23, lines 46-67 through column 24, lines 1-8). However, although Yan et al. teach a URL stored on the non-volatile storage device, Yan et al. fail to explicitly teach an application header stored on the non-volatile storage device, the URL being initialized to access the application body when the loading mechanism detects the application header. Inoue et al. teach a system for loading an application onto a peripheral device from a connected computer network (Inoue et al.: column 2, lines 21-25) similar to that of Yan et al. In addition, Inoue et al. further teach an application header stored on the non-volatile storage device, the URL being initialized to access the application body when the loading mechanism detects the application header (the data file includes a header, where the application information is stored and upon receipt of the data file and based on the header information in the data file, the application from the application server can be downloaded) (Inoue et al.: column 7, lines 1-8 and column 8, lines 10-16). It would have been obvious to one of ordinary skill in the art, having the teachings of Yan et al. and Inoue et al. before him at the time the invention was made, to modify the application loading system taught by Yan et al. to include the application header of Inoue et al. One would have been motivated to make such a combination in order to reduce the burden on a client computer by saving the time and money needed in developing the high capacity of a hard disk or a memory that would have otherwise been necessary in order to continuously store an entire application; by using an application header to store information and downloading the application only according to the

header, the space, memory and management of the client device can be conserved and therefore, the processing and operation efficiency can be improved.

Referring to claim 22, Yan et al. teach a dedicated servlet configured to have settings that are unique to a network based appliance (each peripheral device registers a peripheral profile with a peripheral database containing a configuration applet that which cause the device to work in a predetermined manner) (column 23, lines 46-67).

3. The prior art made of record on form PTO-892 and not relied upon is considered pertinent to applicant's disclosure. Applicant is required under 37 C.F.R. § 1.111(c) to consider these references fully when responding to this action. The documents cited therein teach similar methods for downloading information to peripheral devices.

Response to Arguments

4. Applicant's arguments with respect to claims 1-2, 6-14 and 16-22 have been considered but are moot in view of the new ground(s) of rejection.

5. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ting Zhou whose telephone number is (703) 305-0328. The examiner can normally be reached on Monday - Friday 8:00 am - 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached on (703) 308-3116. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

2 August 2004

CAO (KEVIN) NGUYEN
PRIMARY EXAMINER

